Amendments to the Specification

Please replace paragraph **[0002]** on page 1 of the specification with the following amended paragraph:

[0002] An Appendix containing a computer program listing is submitted on a compact disc, which is incorporated by reference herein in its entirety. The total number of compact discs including duplicates is two. Each of the compact discs includes a file named "anneal.txt", which was created Feb. 7, 2001 and has a size of 19,285 bytes and a file named "swap.txt", which was created Feb. 6, 2001 and has a size of 11,791 bytes.

Please replace paragraph **[0006]** on page 2 of the specification with the following amended paragraph:

[0006] The presentation by Shor (P. W. Shor, "Polynomial-Time Algorithms for Prime Factorisation and Discrete Logarithms on a Quantum Computer," SIAM J. Comp., 26:1484, 1997) of an efficient quantum process for factorization has kindled great interest in developing ~~of~~ software for quantum computing. In particular, a lot of attention has focused on developing methods for efficient computations and error correction in quantum computers.

Please replace paragraph **[0007]** on page 2 of the specification with the following amended paragraph:

[0007] In parallel, great effort ~~as~~ has been invested in developing physical implementations of quantum computers that meet the very stringent requirements needed for the coherent manipulation of quantum information. The first proposals for such quantum computers were based on trapped ions, cavity quantum electrodynamics systems, and NMR spectroscopy. NMR-based implementations of quantum computers have been successful at least for a limited number of qubits. See E. Knill et al., "An Algorithmic Benchmark for Quantum Processing," Nature, 404:368 (2000), which is hereby incorporated by reference in its entirety. However, the inherent limitations of NMR-based quantum computers have motivated the search for more scalable designs.

Please replace paragraph **[0008]** on page 2 of the specification with the following amended paragraph:

[0008] The high level of expertise available in solid-state based technologies establishes solid-state systems as a leading candidate for the realization of a useful (several thousands of qubits) quantum computer. Solid-state quantum computers have been proposed including Josephson junctions, quantum dots, or spin resonance transistors as qubits. Recent experimental success at coherently manipulating a solid-state qubit, for example, as described by Y. Nakamura et al., "Coherent Control of Macroscopic Quantum States in a Single-Cooper Pair Box," Nature (London), 398:786, 1999, gives ~~good confidences~~ confidence that solid-state quantum computers are practical. However, the large ~~numbers~~ number of degrees of freedom in solid-state quantum computers typically cause such quantum computers to suffer from short coherence times. (The coherence time is the time during which a quantum state of the quantum computer can evolve before external influences interfere with the quantum states of the qubits.) To take full advantage of the computational power of a solid-state quantum computer, optimization of the software for the specific quantum hardware will be critical. In particular, optimized software that reduces the total coherence time required for a computation will determine whether the quantum computer can complete a specific calculation.

Please replace paragraph **[0010]** on page 3 of the specification with the following amended paragraph:

[0010] One specific embodiment of this invention is a method for reducing required coherence time for a quantum computation. This is accomplished by constructing a second series of operations from a first series by changing the execution order of the commuting operations in a way ~~maximizing~~ that maximizes the number of operations that can be performed simultaneously. This reduces the time required for a quantum computing device to complete the second series of operations. Changing the execution order of the commuting operations can enable simultaneous execution of operations in the second series and/or can eliminate the need for some swap operations.

Please replace paragraph **[0022]** on page 5 of the specification with the following amended paragraph:

[0022] FIG. 1 illustrates an example of a solid-state multi-qubit register 100 containing a linear array of qubits. Quantum registers such as register 100 are further described in co-owned U.S. patent app. ~~Nos.~~ No. 09/452,749 and United States Patent No. 6,459,097 ~~09/479,336~~, which are hereby incorporated by reference in their entirety. Register 100 includes qubits based on Josephson junctions 130-0 to 130-(N-1) that are at the interfaces between a superconducting bank 110 and respective mesoscopic, superconducting islands 120-0 to 120-(N-1). A d-wave superconductor, for example, a high-Tc superconductor such as YBa.sub$_2$Cu$_3$O$_{7-x}$ or any superconductor, in which the Cooper pairs are in a state with non-zero orbital angular momentum makes up one or both of bank 110 and islands 120-0 to 120-(N-1).

Please replace paragraph **[0023]** on page 5 of the specification with the following amended paragraph:

[0023] A Josephson junction having a d-wave superconductor on one or both sides has ground state current that is doubly degenerate. In particular, the ground state current at each of the Josephson junctions 130-0 to 130-(N-1) is non-zero and either clockwise or counter-clockwise. The ground state current at the ~~ith~~ i^{th} Josephson junction has two basis quantum states ~~|CWi⟩~~ $|CW_i\rangle$ and ~~|CCWi⟩~~ $|CCW_i\rangle$ respectively corresponding to clockwise and counterclockwise currents. Generally, each qubit is in a ground state that is a linear combination of the current states. For example, a state of the first qubit is a combination a*~~|CW0⟩~~ $|CW_0\rangle$ +b*~~|CCW0⟩~~ $|CCW_0\rangle$, where a and b are complex numbers. The two basis states ~~|CWi⟩~~ $|CW_i\rangle$ and ~~|CCWi⟩~~ $|CCW_i\rangle$, for each value i from 0 to (N-1), can be arbitrarily assigned respective binary values 0 and 1.

Please replace paragraph **[0029]** on page 7 of the specification with the following amended paragraph:

[0029] A limitation of a quantum register is the coherence time of the qubits. The coherence time generally indicates the time during which the quantum state of a qubit can

evolve before external factors disrupt the quantum state. A quantum calculation must be completed ~~with~~ within the coherence time. Accordingly, one goal of quantum computing is to minimize the time costs of the process or network that performs a quantum calculation.

Please replace paragraph **[0030]** on page 7 of the specification with the following amended paragraph:

[0030] One useful quantum calculation is a quantum Fourier transform. The quantum Fourier transform is a quantum generalization of the classical Fourier transform. In the field of quantum computing, quantum Fourier transforms are the basis for most ~~know~~ known quantum computations. In particular, quantum Fourier transforms are used in Shor's factorization algorithm (and were developed by Shor for this very purpose). The quantum Fourier transform acts on the state $|N-1,\ldots,0\rangle$ of an N-qubit register as shown in Equation 2.

Equation 2: $$QFT_N : |x\rangle \rightarrow \frac{1}{2^{N/2}} \sum_{y=0}^{2^N-1} e^{2\Pi i \frac{xy}{2^n}} |y\rangle$$

In Equation 2, state $|x\rangle$ and each state $|y\rangle$ has definite current configuration (i.e., correspond to definite N-bit values x and y), and the summation over states $|y\rangle$ is a summation over all definite current states (i.e., all N-bit values y). The quantum Fourier transformation of Equation 2 can be performed on ~~a~~ an N-qubit register 100 using two basic quantum gates: a one-qubit gate A_i acting on the state initially corresponding to the ~~ith~~ i^{th} qubit and a two-qubit gate B_{jk} acting on the states initially corresponding to the ~~jth~~ j^{th} and ~~kth~~ k^{th} qubits.

Please replace paragraph **[0031]** on page 8 of the specification with the following amended paragraph:

[0031] The one qubit gate A_i (also known as the Hadamard transformation) is shown in Equation 3.

Equation 3: $$A_i = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

In Equation 3, the basis vectors for the operator are the two basis states $|CW_i\rangle$ or (1 0) and ~~$|CCWi\rangle$~~ $|CCW_i\rangle$ or (1 0) of the ~~ith~~ i^{th} qubit. The one-qubit gate can be implemented in quantum register 100 as described below.

Please replace paragraph **[0032]** on page 8 of the specification with the following amended paragraph:

[0032] Equation 4 corresponds to the two-qubit gate B_{jk}.

Equation 4: $$B_{jk} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{i\theta_{jk}} \end{pmatrix}$$

In Equation 4, the basis vectors for the operator are the four basis states created from the cross product of the basis states ~~$|CWj\rangle$~~ $|CW_j\rangle$ and ~~$|CCWj\rangle$~~ $|CCW_j\rangle$ of the ~~jth~~ j^{th} qubit with the basis states ~~$|CWk\rangle$~~ $|CW_k\rangle$ and ~~$|CCWk\rangle$~~ $|CCW_k\rangle$ of the ~~kth~~ k^{th} qubit. Coefficients j and k specify the pair of qubits on which to apply the quantum gate B_{jk}. The angle ~~θjk~~ θ_{jk} is defined as $\pi/2^{k-j}$ and so is determined by the "distance" between the qubits on which quantum gate B_{jk} is applied. The two-qubit gate ~~Bjk~~ B_{jk} can be implemented in quantum register 100 as described further below.

Please replace paragraph **[0035]** on page 9 of the specification with the following amended paragraph:

[0035] Optimizing a given computational process generally requires expressing the computational process in terms of the elementary set of gates of the quantum architecture in use. For example, Equations 6, 7, and 8 provide a universal set of operators $X(\theta)$, $Z(\phi)$, and $CP(\zeta)$.

Equation 6: $X(\theta) = e^{\frac{-i\sigma_x\theta}{2}}$

Equation 7: $Z(\phi) = e^{\frac{-i\sigma_z\phi}{2}}$

~~Equation 8: $CP(\varsigma) = e^{\frac{-i\sigma_z\otimes\sigma_z\varsigma}{2}}$~~

Equation 8: $CP(\zeta) = e^{\frac{-i\sigma_z\otimes\sigma_z\zeta}{2}}$

In Equations 6, 7, and 8, σ_x and σ_z are Pauli matrices. Operators X(θ) and Z(ϕ) are single qubit operators and can be indexed according to the qubit operand. Operator CP(ζ) is a two qubit operator and ~~require~~ requires a pair of indices to identify the qubit operands.

Please replace paragraph **[0036]** on page 10 of the specification with the following amended paragraph:

[0036] Operators X(θ), Z(ϕ), and CP(ζ) correspond to the elementary set of gates of many solid-state designs and can be implemented in NMR-based quantum computers. See N. A. Gershenfeld and I. L. Chuang, "Bulk Spin-Resonance Quantum Computation," Science, 275:351, 1997. Implementation of Operators X(θ), Z(ϕ), and CP(ζ) in a quantum register such as illustrated in FIG. 1 is described in the paper of A. Zagoskin and A. Blais, 2000, "Operation of universal gates in a solid-state quantum computer based on clean Josephson junctions between d-wave ~~superconductors'~~ superconductors," Phys. Rev. A 61 :042308 ~~(2000)"~~ which is hereby incorporated by reference in its entirety.

Please replace paragraph **[0037]** on page 10 of the specification with the following amended paragraph:

[0037] The elementary operations X(θ), Z(ϕ), and CP(ζ) implement the gate ~~Bjk~~ B_{jk} (on two adjacent qubits j and k) and the gate **[[s]]** ~~Aj~~ A_j (on qubit j) as indicated in Equations 9 and 10.

Equation 9: $A_j = iZ_j(\pi/2)X_j(\pi/2)Z_j(\pi/2)$

Equation 10: $B_{jk} = e^{i\theta_{k-j+1}} Z_j(\theta_{k-j+1}) Z_k(\theta_{k-j+1}) CP_{jk}(\theta_{k-j+1})$

In Equation 10, the phase $e^{i\theta_{k-j+1}}$ ~~$\exp\{i\theta_{k-j+1}\}$~~ depends on which qubits j and k are the operands of operator ~~Bjk~~ B_{jk} but is independent of the states of qubits j and k. The phase $e^{i\theta_{k-j+1}}$ ~~$\exp\{i\theta_{k-j+1}\}$~~ is thus an unimportant global phase factor and can be ignored.

Please replace paragraph **[0038]** on page 10 of the specification with the following amended paragraph:

[0038] The network of FIG. 2 requires applying gate ~~Bjk~~ B_{jk} to nonadjacent qubits. However, a one-dimensional array of qubits generally limits interaction between qubits to nearest-neighbor couplings, ~~e.g.~~ *e.g.*, in quantum register 100 of FIG. 1, each of SETs 130-1 to 130-(N-1) creates controlled **[[a]]** entanglement between two adjacent qubits. Accordingly, applying gate ~~Bjk~~ B_{jk} to nonadjacent qubits entails swapping recursively the states of adjacent qubits so that the states move to adjacent qubits for interaction.

Please replace paragraph **[0042]** on page 12 of the specification with the following amended paragraph:

[0042] Further, instead of only swapping qubit j toward qubit k, as in FIG. 3A, simultaneously swapping qubits j and k with their neighbors as in FIG. 3B further reduces the number of time steps. Accordingly, the number of time steps used to juxtapose qubit states initially in the ~~jth~~ j^{th} and ~~kth~~ k^{th} qubits and return the interacted states to their original position is reduced to $2\lceil |j-k|/2 \rceil$, where $2\lceil x \rceil$ is the smallest integer larger than x.

Please replace paragraph **[0043]** on page 12 of the specification with the following amended paragraph:

[0043] An additional simplification of the computational process is that states of qubits that have been juxtaposed don't have to be moved back to their original location. Instead, once two qubit states have been brought together and interacted, the next reorganization should be done in a way optimizing the realization of the following quantum gates. The location of the qubit states in the quantum register can be tracked classically, and bits can be ~~reorder~~ reordered as required either during the initialization of qubits or when interpreting results read from the quantum register.

Please replace paragraph **[0044]** on page 12 of the specification with the following amended paragraph:

[0044] The above simplifications of the quantum computation ~~reduces~~ reduce the number of physical time steps from $42(|j-k|-1)$ to $12\lceil |j-k|/2 \rceil$ for a single swap sequence to juxtapose the ~~jth~~ j^{th} and ~~kth~~ k^{th} qubits.

Please replace paragraph **[0046]** on page 13 of the specification with the following amended paragraph:

[0046] The computational process of FIG. 4 can be further optimized, in terms of the number of time steps, using reordering and simultaneous execution of gates that commute. In particular, 1-qubit gates such as gate ~~Ai~~ A_i commute with the swap operation. The gate ~~Ai~~ A_i is performed on the qubit containing the evolved state originally corresponding to the ~~ith~~ i^{th} qubit and accordingly may be performed on a different qubit after a swap. A 1-qubit gate can be performed simultaneously with a swap only if both are not acting on the same qubit. A 2-qubit gate B_{jk} commutes with a swap only if they are not acting on the same qubit(s). For example, in Fig. 4, operation A_2 is during time interval 403 when the evolved state S2 is the state of the qubit Q2. Accordingly, operation A_2 is performed on qubit Q2. A swap operation during time interval 404 swaps state S1 from qubit Q1 into qubit Q2 and swaps state S2 from qubit Q2 into qubit Q1. As a result, states S3 and S1 will be adjacent in the register for operation B_{13}. By changing the order of application of the swap (interval 404) and A_2 (interval 403), A_2 and B_{13} can now be performed simultaneously. This is shown in the intervals 503 and 504 of FIG. 5.

Please replace paragraph **[0047]** on page 13 of the specification with the following amended paragraph:

[0047] Communitivity of other operators ~~permit~~ permits similar time savings. Specifically, as indicated by the commutators in Equations ~~14,15,~~ 14, 15, and 16, gate A_i commutes with gate A_j if i is not equal to j, gate A_i commutes with gate B_{jk} if i is not equal to j or k, and gate B_{jk} commutes with B_{rs} for all j, k, r, and s.

~~Equation 14: [Ai,Aj] = 0 if i ≠ j~~
Equation 14: $[A_i,A_j] = 0$ if $i \neq j$

~~Equation 15: [Ai,Bjk] = 0 if i ≠ j or k~~
Equation 15: $[A_i,B_{jk}] = 0$ if $i \neq j$ or k

~~Equation 16: [Bjk,Brs] = 0 for all j, k, r, and s.~~
Equation 16: $[B_{jk},B_{rs}] = 0$ for all j, k, r, and s.

These commutation relations allow permutations of the order of the logical operations of the computational process of FIG. 4. For example, in FIG. 4, swap operations during interval 408 put the states evolved from states S0 and S3 in adjacent qubits Q1 and Q2 for operation B_{03} during interval 409. A subsequent swap during time interval 410 moves the state evolved from S1 into qubit Q2 for operation B_{01} [[is]] during interval 411 after operations B_{02} and B_{03}. In this ordering of operations, the swap operation of interval 410 undoes one of the swap operations of interval 408.

Please replace paragraph **[0050]** on page 14 of the specification with the following amended paragraph:

[0050] Using this construction of FIG. 6 recursively on the operation network for QFT_{n-1} provides an optimized network for QFT_n. Constructing QFT_n from QFT_{n-1} requires addition of n logical gates A_{n-1}, $B_{(n-2),(n-1)}$, . . . $B_{0,(n-1)}$ and n-1 swaps between qubits n-1 and n-2, qubits n-2 and n-3, . . . and qubits 1 and 0, the swaps being interleaved with the added two qubit operations B. However, the number of added time steps is still 29, and the number of time steps required for this network construction of QFT_n is

8+29(n-2)(≈O(n))for n>2. The number of time steps required for a ~~straitforward~~ straightforward conventional implementation of a network for ~~QFTn~~ QFT_n is $10n-11n^2+4n^3$ ($\approx O(n^3)$). Accordingly, the networks disclosed here provide significant performance gains.

Please replace paragraph **[0051]** on page 15 of the specification with the following amended paragraph:

[0051] FIG. 8 shows the time cost of quantum Fourier transforms as a function of the number of qubits for up to 300 qubits on a logarithmic scale for both improved networks (black circles) and conventional non-improved networks (open squares). The improved networks were obtained numerically using the method disclosed above. In particular, grouping and parallel execution of operations that commute while not acting on similar qubits can maximize performance of parallel operations to minimize the required time for a given network. Such optimizations can be performed efficiently, taking a ~~fews~~ few minutes for a 300 ~~qubits~~ qubit network on a conventional desktop computer. The file "swap.txt" in the CD appendix contains a listing for the program that performs the automated optimization of ~~QFTn~~ QFT_n.

Please replace paragraph **[0052]** on page 15 of the specification with the following amended paragraph:

[0052] For very small networks, both curves in FIG. ~~6~~ 8 coincide while, for larger networks, it is clear that the numerical data correspond to circuits ~~which~~ that are much more efficient. From the logarithmic plot of the numerical data, ~~we obtain~~ a slope of 1.08 ± 0.01 is obtained for networks involving more than 10 qubits. This confirms that the quantum Fourier transform can be implemented in O(n) time steps on n quantum bits. This corresponds to a speedup of order $O(n^2)$ compared to non-optimized networks. Efficient use of swaps and massive (classical) parallelism ~~are~~ is responsible for this speedup. Indeed, speedup by a factor of O(n) can be seen to come from the fact that $O(n^3)$ swaps are necessary in ~~non-optimized~~ non-optimized circuits while only $O(n^2)$ are necessary in the optimized case. The other factor of O(n) comes from the fact that up to n simultaneous operations can be ~~realize~~ realized on n qubits. As in the case of classical parallel computers, this provides a speedup of order O(n).

Please replace paragraph **[0054]** on page 15 of the specification with the following amended paragraph:

[0054] Minimizing the time required for the network corresponds for solving a constrained optimization problem and has many similarities to the problem of placement occurring in VLSI related technologies. In placement, one seeks to arrange the components of a classical circuit in a way ~~minimizing~~ that minimizes the length of interconnecting wires and area of the circuit. Heuristitics like simulated annealing or tabu ~~search~~ searches are known to give good results for such problems. The problem of optimizing a network for a quantum calculation is very similar but with the additional complication that reordering two logical operations at a given location in a circuit will change the sequence and possibly the number of swaps needed at all further points in this circuit. The file "anneal.txt" in the CD appendix contains a listing for a program that performs simulated annealing to optimize a network for a ~~QFTn~~ QFT_n. As this is a heuristic process, the program will not provide optimal solutions but solutions ~~which~~ that are close to the optimal solution.